





Prática de Usabilidade e Navegabilidade de Interface Gráfica

Nessa aula abordaremos o tema Princípios e Métodos do C#, mas para isso iremos percorrer a seguinte trilha pedagógica:

1. Criação de um arquivo de código: saber e entender como inserir um arquivo de código no Unity;



- **Variáveis e Comentários:** aprender as características de variáveis e comentários em C#;



- **Conversões de Dados:** entender a importância de se saber conversão de dados;



- **Operadores:** entender as diferenças e características de operadores em C#.

1. Criação de um arquivo de código



Sabemos que a Unity fornece uma grande quantidade de componentes prontos como é o caso de áudios, física, efeitos de partículas e até mesmo renderização, porém, é importante que você conheça, pelo menos, a lógica de programação.

No Unity você poderá criar a lógica do seu jogo por meio de scripts. Os scripts normalmente comportam-se como componentes na Unity e estão ligados aos GameObjects (ARONOWITZ, 2021, p. 14).

É provável que nessa altura do seu curso, você já saiba um pouco de programação, mas caso você não conheça vamos abordar conteúdos que vão contribuir para que consiga programar seus jogos mais facilmente.

Vamos começar pelo começo? 😄 (risadas)

Iremos iniciar pelo básico, ou seja, aprendendo a criar um arquivo de código no Unity. Para isso:

- Abra o seu projeto Unity (Figura 1);
- Em Project vá na pasta Scripts, se não tiver você deverá criá-la (Figura 1 - a);
- Dentro da pasta clique com o botão direito (Figura 1 - b);
- Agora vá em Create (Figura 1 - c) selecione “C# Script” (Figura 1 - d);
-

Nomeie o arquivo (Figura 2). Exemplo: Variáveis. Operadores etc. Busque nomear com o tipo de atributo que ele representará.



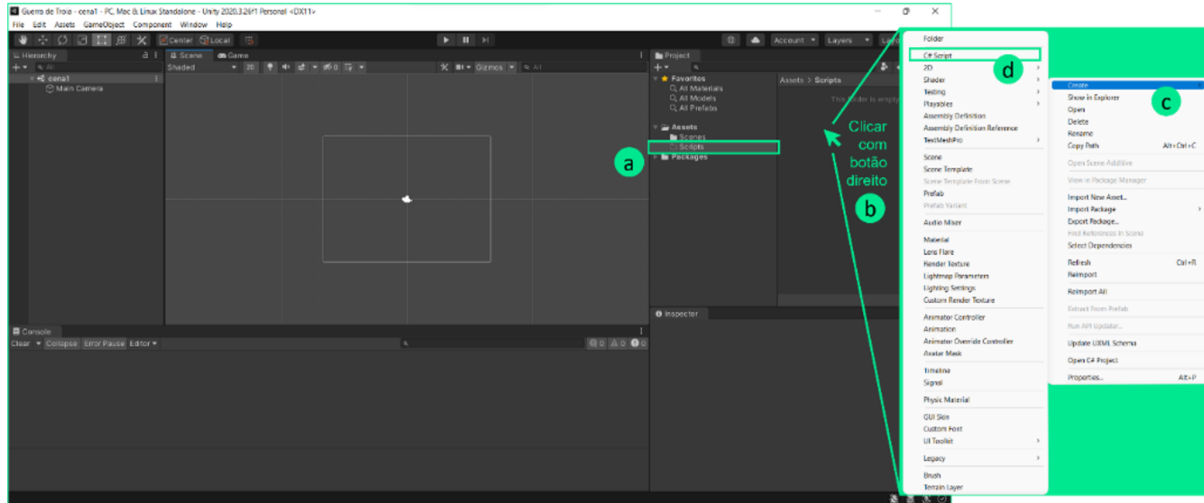


Figura 1 – Criando um arquivo de código | Fonte: Autor, 2022.

O C# pertence ao grupo de linguagem de programação que são orientadas a objetos. Desenvolvido pela Microsoft, ele faz parte da plataforma .NET, uma plataforma que fornece uma vasta gama de bibliotecas de classe, interfaces de programação e utilitários (ARONOWITZ, 2021, p. 14).

2. Variáveis e Comentários

Agora que você criou o arquivo de código em C#, poderá reparar que ao clicar no arquivo, em “Inspector” você conseguirá visualizar alguns códigos que foram adicionados automaticamente no arquivo criado.

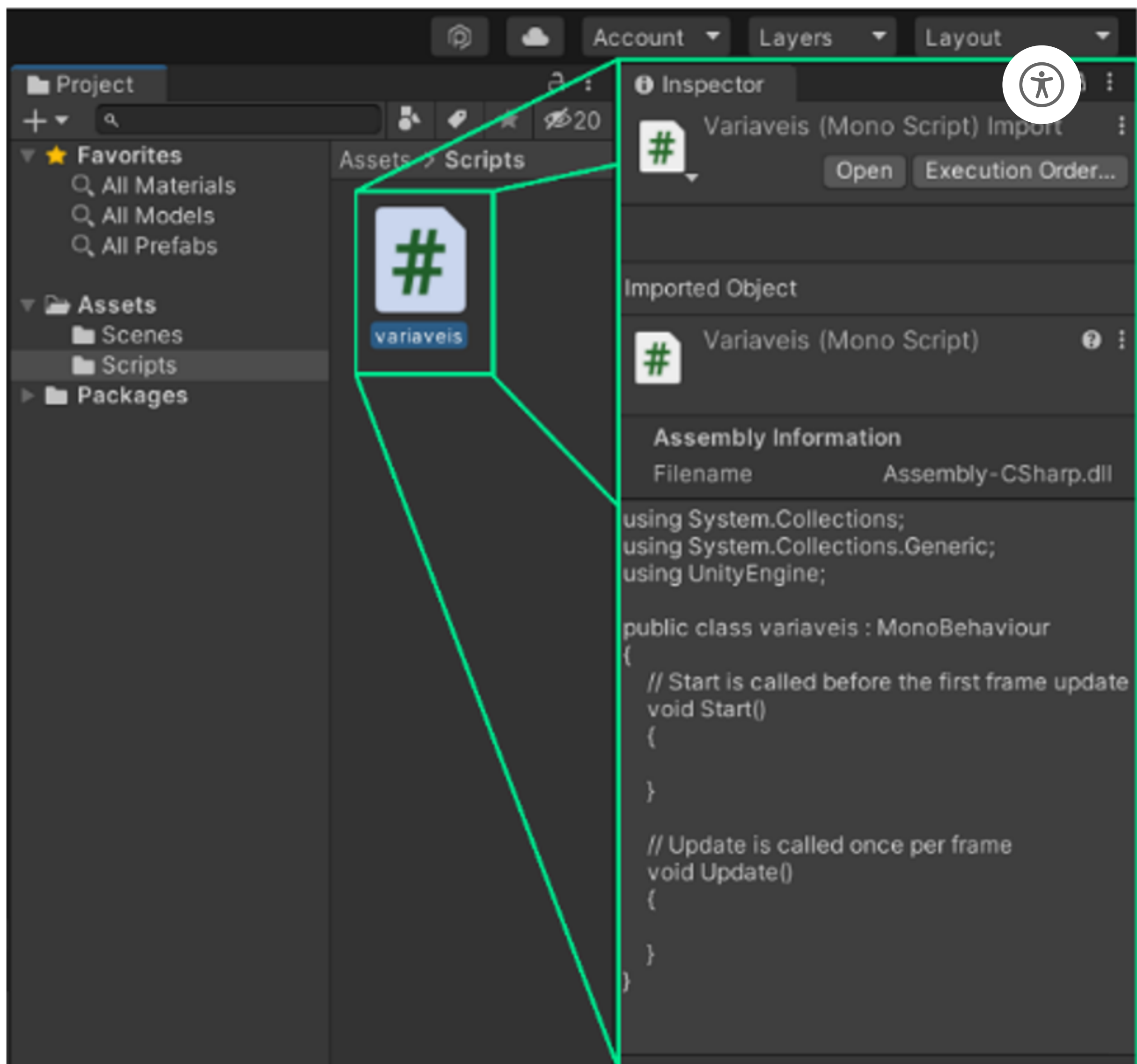


Figura 2 – Arquivo de código | Fonte: Autor, 2022

Quando você clicar duas vezes sobre o arquivo de código criado ele irá abrir uma janela pedindo para você definir o programa que você prefere usar para escrever e/ou editar o código. No meu caso vou usar o Sublime Text (Figura 3) que é um editor de código que funciona nos seguintes sistemas operacionais:

- macOS (é necessário 10.9 ou posterior);
- Windows - também disponível como versão portátil;
- Repositórios Linux - também disponível como tarball x86-64 ou ARM64.

O Sublime Text pode ser baixado e avaliado gratuitamente, no entanto, uma licença deve ser adquirida para uso contínuo.

E:\E-mail\2022\Janeiro\Descomplica\Guerra de Troia\Assets\Scripts\variaveis.cs - Sublime Text (UNREGISTERE

File Edit Selection Find View Goto Tools Project Preferences Help

variaveis.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class variaveis : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

Fonte: Autor, 2022 | Figura 3 – Arquivo de código no Sublime Text

Agora vamos ver como trabalhar com Variáveis e Comentários na linguagem C#. Mas, o que seriam comentários?

Comentários em uma linguagem de programação é importante para organizar e contribuir para o desenvolvimento do código. Para fazer comentários, no C# é bem simples, você pode declarar um comentário de duas formas:

- **// no início do comentário:** para uma única linha. Exemplo:

//comentário do game



2. **Comentário entre /* /:** *para várias linhas de comentário escreva o conteúdo entre esses sinais. Ex:*

/

Olá, mundo! Esse é o comentário de um jogo.

Estamos aprendendo o básico do C# para o desenvolvimento de jogos por com o Unity. Bora programar?

*/

Os comentários serão muito importantes para que você reconheça rapidamente o que o script irá realizar. Use e abuse dos comentários para facilitar o seu trabalho.

E as variáveis? O que são, onde vivem, o que fazem e como sobrevivem?

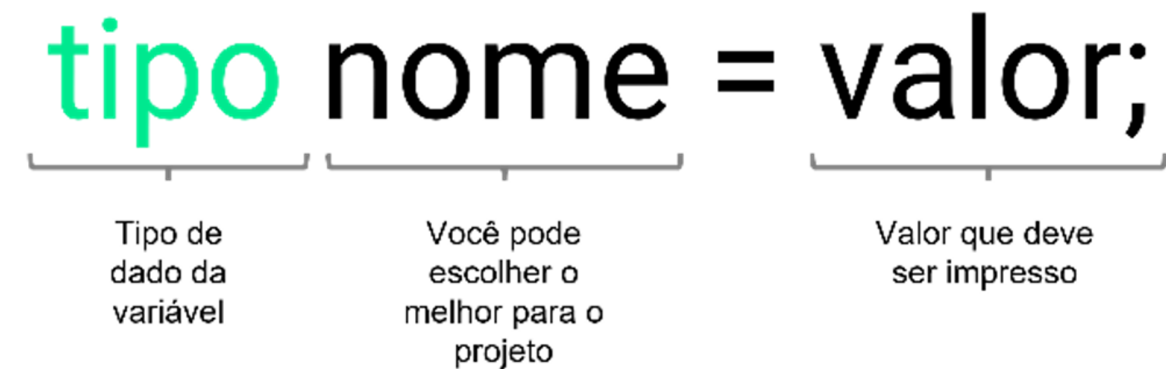
> As variáveis são utilizadas para armazenar informações na memória em tempo de execução da aplicação, isso significa que essas informações estarão disponíveis enquanto a aplicação estiver em execução e, mais precisamente, enquanto a classe ou método em que ela foi criada estiver em execução (DEVMEDIA, 2019).

Mas você deve estar se perguntando: Para que eu irei usar uma variável em um jogo? Quando você estiver criando um jogo e fizer um personagem, por exemplo, ele provavelmente precisará ter algumas características (nome, quantidade de vida, poderes etc.) e é aí que entram as variáveis. Você guardará dentro dessas variáveis esses tipos de informações.

Mas, como se declara uma variável e quais são os tipos de variáveis?



A declaração de uma variável, no C#, acontece por meio da declaração do tipo de dado que a variável vai armazenar e o nome da variável (Figura 4).



string nomePersonagem = "Jeff";

Figura 4 – Declaração de Variável | Fonte: Autor, 2022

Existem diversos tipos de dados que podem ser usados por uma variável, entre eles podemos citar:

Tipo de Valor	Código	Exemplo
VARIÁVEIS PADRÃO		
Números inteiros	int	int moedas = 20;
Flutuantes	float	float VidaPersonagem = 10.5f;
Booleanos	bool	bool PersonagemVivo = true;
Texto	string	string nomePersonagem = “Jeff”;
Atribuição futura	var	var lifePoint; lifePoint = 5;
VARIÁVEIS DO MONOBEHAVIOUR (UNITY)		
Objetos do Jogo	GameObject	GameObject carro;
Transformar objetos	Transform	Transform espada;

Tabela 1 – Tipos de Variáveis | Fonte: AutorD, 2022

Esses dados podem ser alterados em tempo real. Vale ressaltar que as variáveis também podem ser públicas ou privadas, para tanto você precisaria declarar isso no script:

public float Vida Personagem = 10f;

Quando a variável é pública, o dado poderá ser acessado no Editor da Unity e através de outros scripts, caso contrário só será acessada por meio do próprio script.

Agora que você já sabe o que é e como usar uma variável iremos falar sobre métodos. Métodos executam as instruções de um objeto, ou seja, é um bloco de códigos que você irá utilizar para executar instruções, como por exemplo (Figura 5):

- Iniciar algo



- Fazer algo
- Aplicar algo

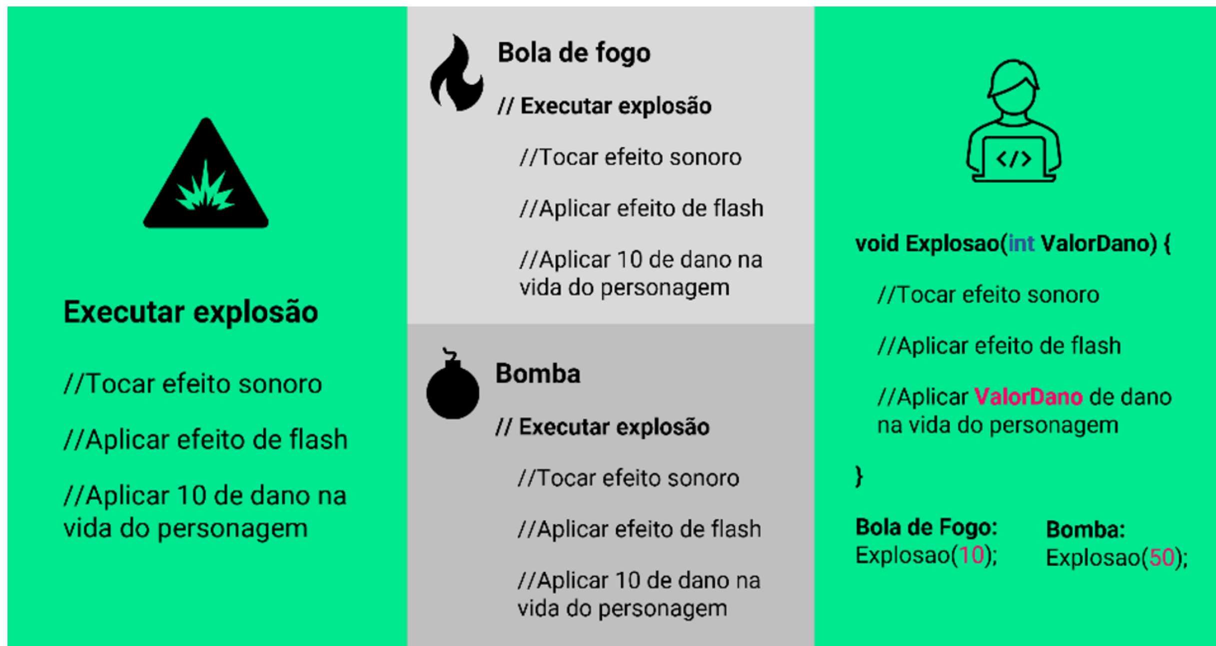


Figura 5 – Exemplo de Método | Fonte: adaptado de SOARES, 2019

3. Conversões de Dados

Agora é a hora de abordarmos o tema Conversão de Dados. É muito comum, em jogos digitais, termos que realizar a conversão de um tipo de informação para outro.

No Menu do Jogo Árida (Figura 6), por exemplo, temos imagens que representam fome e sede que pega informações em algum lugar e apresenta uma fração do inteiro na imagem para dizer se a personagem principal está com algum tipo de deficiência e inclusive se zerar a personagem morre.



Figura 5 – Exemplo de Conversão no Jogo "Arida: Backland's Awakening" | Fonte: Autor, 2022

Um exemplo prático é a conversão de números float para números inteiros. Onde eu poderia criar um valor para uma variante chamada "carteira" como um número inteiro (int), uma outra variável chamada "carteira 2" com número flutuante (float) de "10.5f". Depois na execução converteria o valor flutuante em um número inteiro. Veja o exemplo na Figura 6:

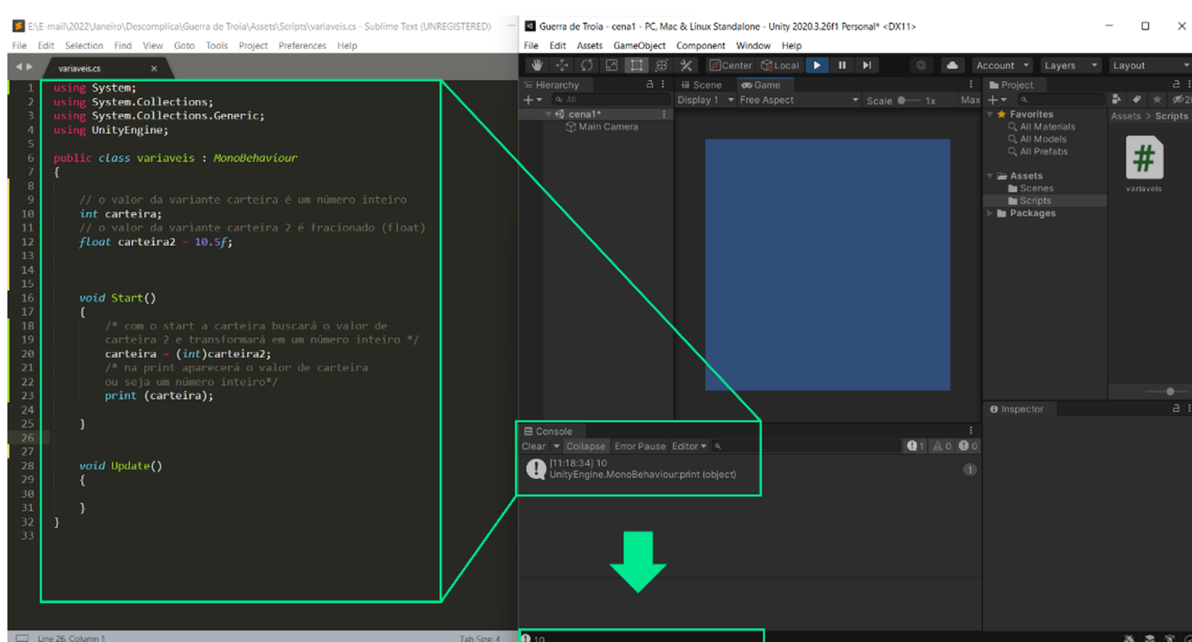


Figura 6 – Exemplo de Conversão Números Flutuantes em Inteiros | Fonte: Autor, 2022.

Podemos acrescentar itens booleanos e até mesmo texto em scripts gerando conversões mais complexas. Se você quiser se aprofundar no assunto te indico ler a Documentação do C# no site da Microsoft <https://docs.microsoft.com/pt-br/dotnet/csharp/>. Nesse site você encontrará os conceitos básicos e tutoriais dessa linguagem orientada a objetos. Existe também o manual de script da Unity <https://docs.unity3d.com/Manual/ScriptingSection.html>, vale ler e pesquisar as possibilidades.

4. Operadores

Vamos falar agora sobre operadores. Os operadores podem ser muito utilizados em Jogos Digitais. Os operadores apresentam um tipo de operação que será executada, gerando assim novos valores a partir de um ou mais operandos. Eles podem ser:

4.1 Operadores Aritméticos

Os Operadores Aritméticos são os operadores matemáticos para realizarmos operações de soma, subtração, multiplicação e divisão (Figura 7).

Operador	Significado
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão)

Operador	Significado
+=	Mais igual
-=	Menos igual
*=	Vezes igual
/=	Dividido igual
%=	Módulo igual

Tabela 2 – Operadores Aritméticos | Fonte: Autor, 2022.

Saiba mais em:

https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/arithmetic-operators

4.2 Operadores Relacionais

Os Operadores Relacionais comparam 2 valores e retornam um valor booleano, ou seja, verdadeiro ou falso. (Figura 8). Entre os operadores podemos citar:

Operador	Significado
==	Igualdade
!=	Diferente
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Tabela 3 – Operadores Relacionais | Fonte: Autor, 2022

4.3 Operadores Lógicos

Os Operadores Lógicos trabalham sobre valores booleanos, ou seja, por meio de verdadeiro ou falso. Seu resultado também será booleano.

Operador	Significado
&&	And = e
||	Or = ou
!	Not = não

Tabela 4 – Operadores Lógicos | Fonte: Autor, 2022

4.4 Operadores Ternários

O operador ternário é composto por três operandos separados pelos sinais "?" e ":" e tem o objetivo de atribuir o valor a uma variável de acordo com o resultado de um teste lógico (DEVMEDIA, 2019-b).

Para entender bem os operadores ternários é importante saber usar um Condicional "if, else e/ou else if". Por exemplo, se um jogador estiver com pouca vida e quiser comprar mais uma, seria necessário que a quantidade de moeda fosse maior que o valor da vida (figura 7).

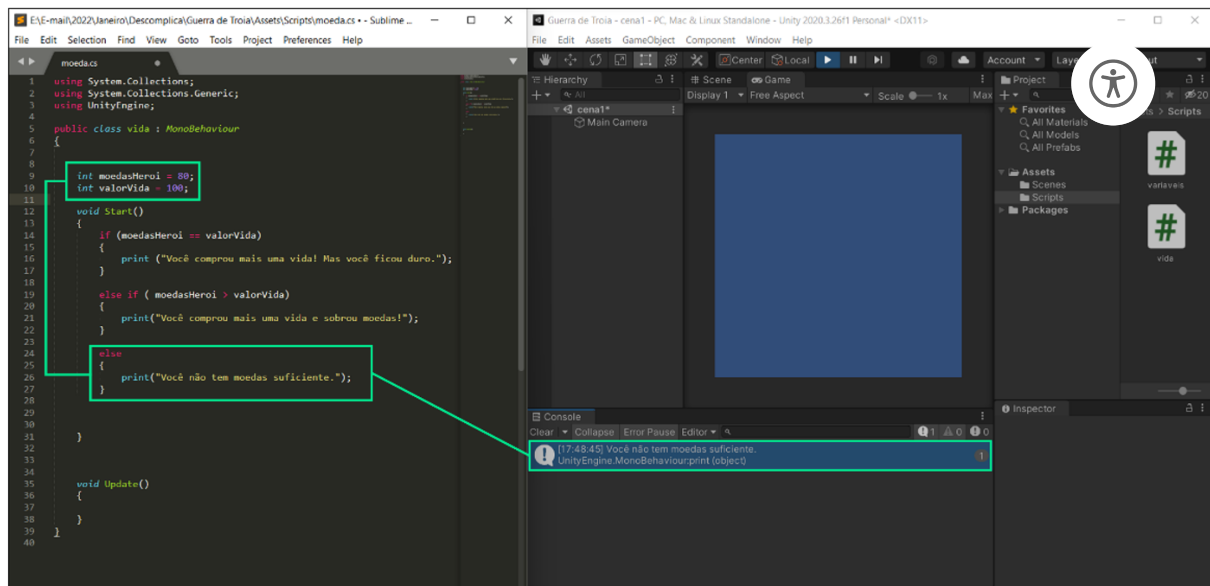


Figura 7 – Exemplo de condicional | Fonte: Autor, 2022

Perceba que a sintaxe para se declarar uma condicional pode variar de acordo com o tipo de condição:

if (variável 1 operador variável 2) { print ("texto"); }

else if (variável1 operador variável 2) { print ("texto"); }

else { print ("texto"); }

Você poderia obter o mesmo resultado usando Operadores Ternários, conforme você poderá verificar na Figura 8.

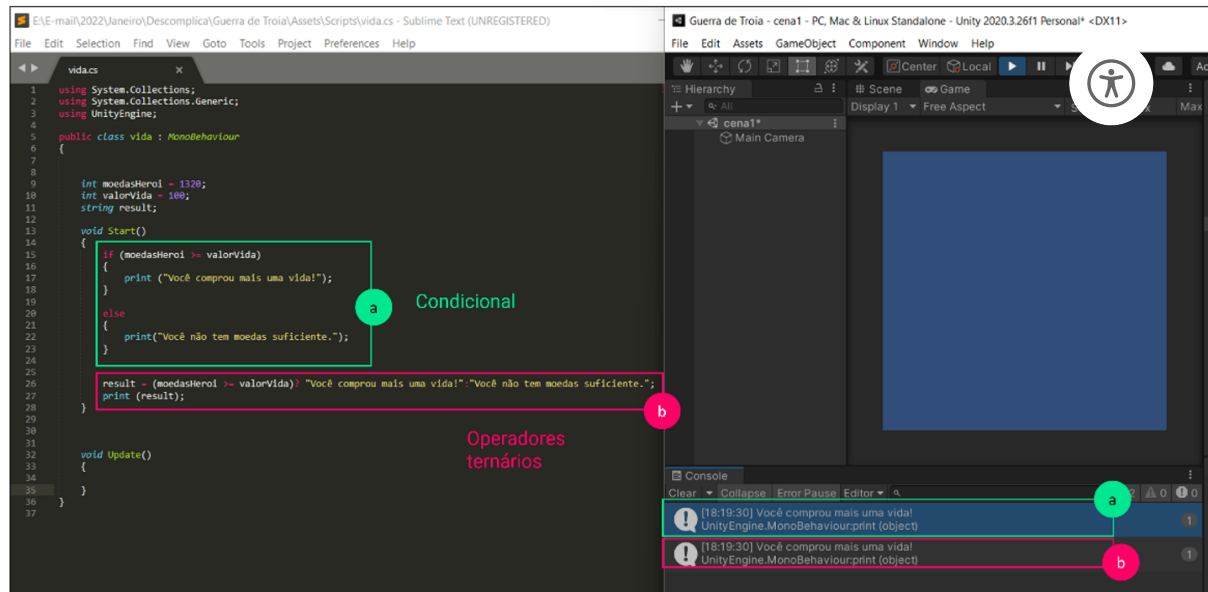


Figura 8 – Exemplo de Operadores Ternários | Fonte: Autor, 2022

Espero que tenham gostado e até mais! Vida longa e próspera!

Atividade Extra

Leia o capítulo do Guia do C# da Microsoft "Operadores e expressões do c# (referência C#)" Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/>

Referência Bibliográfica

ARAÚJO, J. **Tipos de Variáveis no C#.** DEVMedia, 2010. Disponível em: < https://www.devmedia.com.br/tipos-de-variaveis-no-csharp/1677

ARONOWITZ, A. **Programação C# com Unity 3d: Desenho e programação de jogos eletrônicos com o Unity.** Portugal: NLN, 2021. Edição do Kindle.

DEVMEDIA. **Introdução a Variáveis e Constantes no C#.** 2019. Disponível em: < https://www.devmedia.com.br/introducao-a-variaveis-e-constantes-no-csharp/29629>.

___. **Tipos de Operadores do C#.** 2019-b. Disponível em: <https://www.devmedia.com.br/tipos-de-operadores-do-csharp/18873>.

FILHO, C. **Linguagem de programação para Games**: 6 Melhores Linguagens de Programação Para Desenvolver Jogos. DIO, 2021. Disponível em: <https://www.dio.me/articles/linguagem-de-programacao-para-games>.

SOARES, W. **Curso Introdução à programação de jogos em C#.** CSJ, 2019. Disponível em: <https://www.youtube.com/watch?v=QbeUIT_MuZ0>.

Ir para exercício